Report on COMP 3100, Semester 2, 2015

Course Title. Tech Launcher

Course Convenor. Shayne Flynt

Number of Students. 36 (COMP 3100) and 28 (COMP 4500)

Data Gathering To provide the context for the review, the following sources of information were consulted:

- Course: COMP3100 Software Engineering Group Project
- Programs and Courses page: http://programsandcourses.anu.edu.au/course/comp3100

Shayne also provided a series of notes and access to all TechLauncher systems via Dirk Pattinson.

It is clear that the course has not reached 'steady state', with a number of on-going tweaks, some of which (I understand) make parts of this review outdated.

The course is co-badged with a number of other courses in CS, all with similar entries on Programs & Courses.

Class Structure Programs and Courses (P&C) (http://programsandcourses.anu.edu.au/course/comp3100) indicates that the course involves lectures (2 hours per week) and group project time (260 hours).

It is clear that the emphasis in this course is on the group project process, and almost all formal and informal learning takes place in this context.

Learning outcomes P&C list 11 learning outcomes for the course (note: a list of 10-12 outcomes appear to be similar across all the co-badged courses).

Assessment P&C indicates three major assessment items:

- individual portfolio (30
• group project reviews (60
• final group poster (10

This weighting appears to be different for the 2016 course guide (https://anutechlauncher.net/projects/course-guide-2016-s1/wiki/2016_Semester_1_Course_Guide), with no portfolio, and instead a ‘tutor review’ based on individual’s efforts within the group.

Mapping LOs to Assessment The Rubrics for the assessment are in Wattle(https://wattlecourses.anu.edu.au/mod/folder/view.php?id=856554). The LOs are in line with the criteria in the marking rubrics, but are not explicitly mapped.

Based on the available material, almost all of the assessment is based around the group work, both processes within the project and artefacts produced throughout the project.

Feedback Feedback to students appears to be embedded through the group project itself, where regular reviews are conducted within the team and with a tutor.

Feedback for the course is collected through a survey in Wattle with reasonably high completion (86 responses for all co-badged variants). The feedback in this process is promising, with 70% agreement with the satisfaction of the project, team and tutor meetings. This less formal path appears very positive compared to the SELS process.

The SELS reports appear to be very noisy, with little consistency in the time series reports between the differently badged courses. There are also large differences between semesters. It is not clear what the reasons for this might be without understanding the historical context for the course (which I do not).

Cohort There are clear differences in the ‘overall satisfaction’ levels between cohorts in the course, based on the SELS. A brief summary is shown below:

<table>
<thead>
<tr>
<th>Course</th>
<th>2014 S1</th>
<th>2014 S2</th>
<th>2015 S1</th>
<th>2015 S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMP3100</td>
<td>58%</td>
<td>33%</td>
<td>43%</td>
<td>44%</td>
</tr>
<tr>
<td>COMP3500</td>
<td>60%</td>
<td>100%</td>
<td>40%</td>
<td>83%</td>
</tr>
<tr>
<td>COMP3550</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>COMP4500</td>
<td>-</td>
<td>-</td>
<td>38%</td>
<td>33%</td>
</tr>
<tr>
<td>COMP8715</td>
<td>-</td>
<td>75%</td>
<td>-</td>
<td>82%</td>
</tr>
</tbody>
</table>

The SELS indicate that there are 36 students in 3100 in 2015, which makes up (I think) a little over one-quarter of the entire cohort. Beyond that, it is unclear what the make-up of the cohort is.

This could reasonably lead to an assumption that different cohorts of students value the course in different ways.
Summary of convener’s notes Shayne provided comments about the problems and solutions he perceives in the course. These include:

- concerns over student preparedness for the course, including students who are concurrently enrolled in prerequisites
- student concerns over assessment, including peer assessment
- tutors with industry experience are more useful than those without industry experience
- difficulty getting students to make connections to what they have learnt in other courses
- students not understanding course requirements, or being mislead by inaccurate information provided by peers

Kerry agreed with Shayne’s account.

Shayne provided example feedback produced by peers for a regular review, worth 3 marks. The feedback is a highly-developed, clear and comprehensive template for collecting and disseminating feedback in a group project.

A Grading Guidelines document was also provided. This document gives some insight into the philosophy of the course, and a clear discussion about the approach for marking aligned with educational philosophy.

Review Topics Below are comments against the suggested review topics.

1. Is the topic of the course relevant and timely for students?
   TechLauncher could be classified as a ‘capstone’ course, where apply the knowledge they have gained through their degree to a problem of their choosing. A third-year ‘capstone’ (COMP3100) is, perhaps, early in the program, but appears to be appropriate given the main goal is to work in teams to develop a project, which is an activity that requires scaffolded practice.
   Shayne points out that the timeliness may be a significant issue for students, as some students are enrolling in this course at the same time as the first-year prerequisites.

   Group project work for software engineers is clearly a valuable skill to develop in an undergraduate degree. However, problems can manifest in small groups when all the assessment relies on the success of the group.
   From the ‘outside’, there are a few considerations that may improve the apparent preparedness of students for learning in an open-ended project:
Program level
provide more opportunities for students to engage in meaning-
ful group work in first and second year. Developing software is
(typically) not an individual enterprise, and if working in a de-
velopment team is an important graduate skill it should be scaf-
folded earlier in the program (for example, smaller tasks such as
debugging code or adding a module to existing software).

Course level
one of the frustrations from Shayne’s comments is that some stu-
dents do not have the requisite coding skills to participate pro-
ductively in the course. This is not a unique problem, and this
problem is replicated across many degrees and courses in higher
education. However, the ability to write code is not explicitly
stated in the prerequisites (it is assumed), and there are no learn-
ing outcomes or even assessment tasks that explicitly require a
student to be able to write code. If being able to code is a de-
sired outcome of the course, it should be explicitly stated as a
formal prerequisite (and an informal requirement depending on
projects).

Team level
teams are formed around projects of interest based on an estima-
tion of the required roles for the project. Given that diversity of
the skills within a team is desired, it follows that there is a value
that each individual brings to the team. For example, if project
planning is a required role in the team, there is no need for the
project planner to also demonstrate their capacity to code, ex-
cept to the extent that she can communicate effectively with the
software developer so that she can best estimate the amount of
time required for a task.

2. How is it embedded into the overall education offering?
COMP3100 makes up 12 units in the Software Development ma-
jor, and is part of the third-year pattern the Bachelor of Advanced
Computing (Honours).
It is a considerable challenge that students enrolled in different
courses and at different stages of their program and learning are
bundled in together. This could be seen as an opportunity to
promote the diversity required to make successful group projects
work, such as assigning special roles in teams to differently badged
students (noting that this is likely to be already in operation
where appropriate at the team level).
Rather than seeing the assumed pre-requisites as a constraint,
there may actually be some value in removing all specific pre-
requisites, and instead require students to have completed a cer-
tain number of units (eg 96 units, or 24 COMP units). Taking this more open approach could also make the TechLauncher project more attractive to students from outside computer science.

It is unclear why there are so many co-badged variants. In my opinion, all this serves to do is cause difficulty in navigating course information. As far as I could tell and for all intents and purposes, students in differently badged courses are not treated or assessed differently.

Each of the co-badged courses has a very long list of very specific learning outcomes.

3. In which way is feedback given? How timely is the feedback?
The course offers many opportunities for students to get feedback, primarily from their peers or those working close to the project, such as tutors.

The framework for developing feedback is exemplary, and in my opinion demonstrates a high-quality process for group work.

It is not clear how quickly feedback from assessment items is returned. Based on the philosophy of the course, it would appear that the emphasis is on formative feedback rather than any summative feedback. This approach to feedback appears entirely reasonable for the course as it stands.

4. Were all required resources available at all times?
All of the resources required appear to be available as required. The resources, however, require deliberate navigation between multiple systems. Considerable effort has gone into developing these resources.

Whilst there is a great amount of resources to navigate the course, I could not find content resources that might help students approach their project. For example, procedures/frameworks for debugging code or developing customer requirements. I imagine that navigating approaches and concerns is done informally through review sessions and tutorials.

5. Does the mode of delivery foster the learning outcomes?
The mode of delivery appears to be entirely appropriate for the goal and philosophy of the course. However, in my opinion, there are too many learning outcomes to be able to connect to the course delivery or assessment in meaningful ways.

I would suggest that the learning outcomes could be condensed to consider 5 key outcomes (below, with the current LOs as sub-points). This smaller list could simplify the communication of the course objectives, and provide an easier base for constructive alignment around the course assessment and activities.
Effective team work

- Work as an effective member of a team to implement a software based solution that delivers measurable value to an industry or university client.
- Exhibit an awareness of team formation strategies and stages leading to the development of high performing, self-managing teams; sound meeting practice; and how personality traits can impact upon team performance and how to use individual traits to achieve the most from team work.

Clear communication

- Communicate effectively, orally and in writing, with peers, supervisors and commercial clients/stakeholders.
- Participate in a group presentation, including a demonstration, to an audience of peers, clients and supervisors.

Sound decision-making skills

- Make and defend sound engineering decisions.
- Creatively identify and implement a solution to a complex problem that exists within the domain of ICT.

Professional practice skills

- Explain the role and importance of project management, configuration and risk management processes when undertaking a software development project. Demonstrate experience in undertaking the activities associated with these.
- Explain and understand the importance of the different stages of, and activities associated with each, the software development lifecycle (SDLC). Demonstrate experience in all stages of the SDLC.
- Explain the role and importance of standards in software development. Demonstrate experience in tailoring those standards appropriately according to the project they are currently undertaking.

Capacity to act on feedback and personal reflection

- Participate effectively in project and artefact reviews with peers, supervisors and clients/stakeholders.
- Develop life-long learning through reflection, as demonstrated through continual reflection on the software development lifecycle and team work processes experienced throughout the year.

6. Are the learning outcomes covered by the assessment, and are only learning outcomes being assessed?
The rubrics and marking guides cover the majority of the learning outcomes. I found it difficult to connect assessment to some of the LOs, as some LOs describe tasks undertaken in the course rather than learning outcomes (for example, group formation strategies). With the detailed list of LOs and the group-based assessment, I do not believe that it is possible identify an individual’s capacity to meet the LOs.

Notes and Recommendations. COMP3100 clearly has low overall satisfaction in the SELS. However, feedback provided in informal paths such as Wattle for all TechLauncher cohorts suggests that satisfaction is in fact reasonable at \( \sim 70\% \).

Having navigated through the various systems and course structure, there are a number of notable areas of strength as well as areas for improvement.

Noteworthy

- the established methodology for continuous peer assessment is highly developed and well-considered. This should be recognised at a college level as a problem that occurs across all group work, and this approach could inform a methodology for widespread adoption across all professional group-work courses in CS and engineering.
- the real-world context of the course is ambitious and leverages a broad network of professionals for the benefit of students
- the information, structure and philosophy of the course is detailed and considered

Recommendations

As always, there are opportunities for improvement. These recommendations are provided with little understanding of the history or politics of the course and degree program. In no order:

At a program level:

- *foster a community of professional practice* - students shouldn’t have to wait until third-year to create meaningful software. An emphasis should be placed on developing professional practice throughout the degree, from first year

- *streamlined peer evaluation* - the peer evaluation process is automated, but requires offline data entry for the user. This appears to be a major reason for missing 3 marks on a regular basis. A modern, easy-to-navigate peer evaluation system/application would produce better feedback
• simplify enrolment options - the TechLauncher project should have a single course code (perhaps with postgraduate variation) that can be recognised across degree programs and throughout the university

At a course level:

• eliminate specific prerequisites - as each project needs and roles are different, the requirement to have completed specific units appears strange. Relaxing the prerequisites to a number of units (eg 96 units) may be an opportunity to walk-the-talk when it comes to valuing diversity in professional practice

• simplify learning outcomes - the learning outcomes should be simplified to 5-6 thematic outcomes. A suggested grouping is provided in 5 of the Review Topics

• align simple learning outcomes to assessment criteria - the marking criteria for each assessment item should derive from the learning outcomes

• individual assessment - there is no individual assessment in this course, raising the stakes of the group project. I would recommend designing a task, ~ 30% of the final mark, that allows individuals to showcase their achievement of the learning outcomes of the course in a purposeful way. One way to frame this may be to use only the experience of the course to write a job ’application’, addressing the LOs as the selection criteria. In addition, an exit interview with each student to go through their ’application’ would enable a more thorough examination of individual learning.

• regular, strategic communication - as there are no lectures in the course, there are no opportunities to make whole-of-course announcements. Most of the information about the course is provided in self-service mode. A ’newsletter’-style email/post with what’s happening in the coming/following week could orientate students to the immediate requirements.

• develop online learning resources - a field guide, alongside a repository of worked examples and case studies could help orientate students to the expectations of practice in the course. These resources should address relevant topics, such as ’tips for how to run a meeting’, or ’how to have difficult conversations’, etc, and could be used as reference material.

• allow students to focus on strengths?